

◎ 日本国特許庁 (JP)

◎ 特許出願公開

## ◎ 公開特許公報 (A) 平2-245829

◎ Int. Cl.<sup>5</sup>G 06 F 9/46  
15/16

識別記号

3 6 0 B  
4 3 0

庁内整理番号

8045-5B  
6745-5B

◎ 公開 平成2年(1990)10月1日

審査請求 未請求 請求項の数 4 (全9頁)

◎ 発明の名称 マルチタスキング方法

◎ 特 願 平1-65230

◎ 出 願 平1(1989)3月20日

◎ 発 明 者 北 井 克 佳 東京都国分寺市東恋ヶ塚1丁目280番地 株式会社日立製作所中央研究所内

◎ 発 明 者 田 中 義 一 東京都国分寺市東恋ヶ塚1丁目280番地 株式会社日立製作所中央研究所内

◎ 発 明 者 玉 置 由 子 東京都国分寺市東恋ヶ塚1丁目280番地 株式会社日立製作所中央研究所内

◎ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目5番地

◎ 代 理 人 弁理士 中村 純之助

要 約

## 1. 発明の名称

マルチタスキング方法

## 2. 特許請求の範囲

1. 主記憶装置を共有し、複数の処理装置でタスクの子孫であるサブタスクと成る手続きを含むプログラムを処理する多重処理システムのマルチタスキング方法において、

手続きの実行が可能か否かを示す実行可否情報と、サブタスクとなる手続きを示す情報を含み、手続きの実行を管理する制御ブロックを演算器作成し、上記プログラムの処理開始時に上記演算器の制御ブロックのそれぞれの実行可否情報を実行不可とし、上記プログラムの実行時にサブタスクを生成する要求があったとき上記演算器の制御ブロックから実行不可の実行可否情報を持つものを選択し、かつ実行可否情報を実行可能を示す情報にセットすることによりサブタスクを生成し、サブタスクを消滅する要

求があったとき対応する制御ブロックの実行可否情報を実行可能としサブタスクを消滅するマルチタスキング方法。

2. 請求項第1記載のマルチタスキング方法において、

上記プログラムの実行開始時に作成する制御ブロックの数と、上記プログラムの実行手順を記述するジョブ制御言語で指定するタスクの最大値とすることを特徴とするマルチタスキング方法。

3. 主記憶装置を共有し、複数の処理装置でサブタスクと成る手続きを含むプログラムを処理する多重処理システムのマルチタスキング方法において、手続きの実行が可能か否かを示す実行可否情報と、タスクとなる手続きを示す情報を含み、手続きの実行を管理する制御ブロックを演算器作成し、

上記プログラムの処理実行時に、手続きの実行が可能か否かを示す実行可否情報と、タスクとなる手続きを示す情報を含み、手続きの実行

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 02-245829

(43)Date of publication of application : 01.10.1990

(51)Int.Cl.

G06F 9/46

G06F 15/16

(21)Application number : 01-086200

(71)Applicant : HITACHI LTD

(22)Date of filing : 20.03.1989

(72)Inventor : KITAI KATSUYOSHI  
TANAKA GIICHI  
TAMAOKI YOSHIKO

## (54) MULTI-TASKING METHOD

## (57)Abstract:

PURPOSE: To attain efficient multi-tasking by re-using the same logical task for different procedures.

CONSTITUTION: One control block showing the impossibility of execution is selected first among plural control blocks 6a-6g by a request for the generation of a sub-task at the time of executing a program. Information showing the entrance point of the procedures 7a-7g of the sub-task are set in the selected control block, and information showing the possibility of execution is set so as to generate the sub-task. Information showing the impossibility of execution is set in the control block of the sub-task by the request for deleting the sub-task, and therefore the sub-task is deleted. After the deletion of the sub-task, the control blocks 6a-6g are re-used and the other sub-task can be generated. Thus, the number of the control blocks can be limited for re-using the control blocks 6a-6g. An overhead for generating a slave-task is reduced, and a program processing can be speeded up.

